

Задача А. Кто прочитал тот сдохнет

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Однажды вам приснился кошмар. Вы проснулись и записали условие задачи из сна:
Вам дано подвешенное дерево на n вершинах, у каждой вершины v есть свой целый вес w_v .
Далее вам поступают m запросов одного из следующих типов:

- $0\ x\ y$ – присвоить всем вершинам в поддереве вершины x вес y .
- $1\ x$ – поменять корень на вершину x .
- $2\ x\ y\ z$ – присвоить всем вершинам на пути от вершины x до вершины y вес z .
- $3\ x$ – вывести минимум в поддереве вершины x .
- $4\ x$ – вывести максимум в поддереве вершины x .
- $5\ x\ y$ – увеличить вес всех вершин в поддереве вершины x на y .
- $6\ x\ y\ z$ – увеличить вес всех вершин на пути от x до y на z .
- $7\ x\ y$ – вывести минимум на пути от вершины x до вершины y .
- $8\ x\ y$ – вывести максимум на пути от вершины x до вершины y .
- $9\ x\ y$ – изменить предка вершины x на вершину y , если y находится в поддереве вершины x ничего не делать не надо.
- $10\ x\ y$ – вывести сумму на пути от x до y .
- $11\ x$ – вывести сумму в поддереве вершины x .

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 10^5$) — число вершин и число запросов соответственно.

В следующих $n - 1$ строках содержатся по два целых числа, описывающих ребра.

В следующих n строках содержатся изначальные веса вершин.

Следующая строка содержит корень дерева.

Следующие m строк содержат запросы в описанном выше формате. Все вершины даны в 1-нумерации. Гарантируется, что все промежуточные вычисления помещаются в `int32_t`.

Формат выходных данных

Выведите ответы на все запросы, по одному числу в строчке.

Примеры

стандартный ввод	стандартный вывод
5 5 2 1 3 1 4 1 5 2 4 1 4 1 2 1 10 2 3 3 1 7 3 4 6 3 3 2 9 5 1	9 1 1
10 12 2 1 3 2 4 2 5 3 6 4 7 5 8 2 9 4 10 9 791 868 505 658 860 623 393 717 410 173 4 0 8 800 1 4 2 8 2 103 3 9 4 4 5 7 304 6 8 8 410 7 10 8 8 1 8 9 6 9 10 2 3 11 5	173 860 103 791 608 1557

Задача В. Переворот

Имя входного файла: `reverse.in`
Имя выходного файла: `reverse.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан массив. Надо научиться обрабатывать два типа запросов.

- 1 L R - перевернуть отрезок [L, R]
- 2 L R - найти минимум на отрезке [L, R]

Формат входных данных

Первая строка файла содержит два числа n, m . ($1 \leq n, m \leq 10^5$) Во второй строке находится n чисел a_i ($1 \leq a_i \leq 10^9$)- исходный массив. Остальные m строк содержат запросы, в формате описанном в условии. Для чисел L,R выполняется ограничение ($1 \leq L \leq R \leq n$).

Формат выходных данных

На каждый запрос типа 2, во входной файл выведите ответ на него, в отдельной строке.

Примеры

reverse.in	reverse.out
10 7	3
5 3 2 3 12 6 7 5 10 12	2
2 4 9	2
1 4 6	2
2 1 8	
1 1 8	
1 8 9	
2 1 7	
2 3 6	

Задача С. Динамический Лес

Имя входного файла: `linkcut.in`
Имя выходного файла: `linkcut.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам нужно научиться обрабатывать 3 типа запросов:

1. Добавить ребро в граф (`link`).
2. Удалить ребро из графа (`cut`).
3. По двум вершинам a и b вернуть длину пути между ними (или -1 , если они лежат в разных компонентах связности) (`get`).

Изначально граф пустой (содержит N вершин, не содержит ребер). Гарантируется, что в любой момент времени граф является лесом. При добавлении ребра гарантируется, что его сейчас в графе нет. При удалении ребра гарантируется, что оно уже добавлено.

Формат входных данных

Числа N и M ($1 \leq N \leq 10^5 + 1$, $1 \leq M \leq 10^5$) — количество вершин в дереве и, соответственно, запросов. Далее M строк, в каждой строке команда (`link` или `cut`, или `get`) и 2 числа от 1 до N — номера вершин в запросе.

Формат выходных данных

В выходной файл для каждого запроса `get` выведите одно число — расстояние между вершинами, или -1 , если они лежат в разных компонентах связности.

Примеры

linkcut.in	linkcut.out
3 7 get 1 2 link 1 2 get 1 2 cut 1 2 get 1 2 link 1 2 get 1 2	-1 1 -1 1
5 10 link 1 2 link 2 3 link 4 3 cut 3 4 get 1 2 get 1 3 get 1 4 get 2 3 get 2 4 get 3 4	1 2 -1 1 -1 -1

Задача D. Connect

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 256 мегабайт

Дан список из m ребер, пронумерованных от 1 до m . Посчитайте количество подотрезков $[l, r]$ ($1 \leq l \leq r \leq m$) таких, что граф на n вершинах, содержащий только ребра из этого подотрезка, является связным.

Формат входных данных

Первая строка входных данных содержит два числа n и m ($2 \leq n \leq 50000$, $1 \leq m \leq 200000$) — число вершин и ребер. Следующие m строк содержат описания ребер. На $i + 1$ -й строке даны два целых числа u_i, v_i — концы i -го ребра ($1 \leq u_i, v_i \leq n$).

Формат выходных данных

Выведите ответ на задачу.

Пример

стандартный ввод	стандартный вывод
4 4 1 2 2 4 1 3 1 4	3

Задача Е. Кратчайший путь

Имя входного файла:	<code>shortest.in</code>
Имя выходного файла:	<code>shortest.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В некотором царстве, в некотором государстве есть несколько городов, соединённых двусторонними дорогами. При этом страна разбита на провинции, внутри каждой провинции можно добраться от любого города до любого другого, а между провинциями пути не существует. Внутри каждой провинции есть один выделенный город, который называется столицей этой провинции.

Компания «AloeVera» занимается пассажирскими перевозками в этом государстве. Компания перевозит людей от одного города до другого по кратчайшему пути. Однако, дорожная сеть постоянно развивается, поэтому периодически в государстве появляются новые дороги. Строительство новой дороги — очень важное событие, поэтому при появлении новой дороги все стремятся проехать именно по ней. Более точно, при добавлении новой дороги между городами u и v существует две возможности:

1. Города u и v находятся в разных провинциях. Тогда после добавления этой дороги эти две провинции объединяются, при этом столицей становится столица первой провинции (той, в которой до добавления находилась вершина u).
2. Города u и v находятся в одной провинции. Тогда происходит перераспределение транспортных потоков. Услышав об открытии новой дороги, все продолжают добираться из столицы в другие города провинции по кратчайшему пути, но в случае, если таких путей существует несколько, то предпочтение отдается тому из них, на котором находится новая построенная дорога. После этого, если найдутся дороги, по которым никто ездить не будет, они навсегда удаляются из дорожной сети. Будьте внимательны: может так случиться, что по новой дороге никто ездить не будет, и тогда её не нужно добавлять вообще.

Иногда в государстве некоторые дороги приходят в негодность. При этом, если после удаления этой дороги некоторая часть провинции становится недостижимой из столицы, то она становится независимой провинцией, и её столицей назначается город, находящийся в этой провинции и являющийся концом удалённой дороги.

Основная же задача компании — сообщать людям о кратчайшем расстоянии между двумя какими-то городами. Ваша задача состоит в том, чтобы автоматизировать работу «AloeVera». При этом люди, сообщающие о разрушении дорог, иногда запаздывают или оперируют неверной информацией, и сообщают о разрушении дороги, которая уже была разрушена или вообще никогда не существовала.

Для решения поставленной задачи компания «AloeVera» использует следующий интерфейс:

- 1 u v — запрос об удалении дороги между городами u и v . Будьте внимательны, эта дорога могла быть разрушена ранее, или вообще никогда не существовать!
- 2 u v w — запрос о добавлении новой дороги между городами u и v , время проезда по которой будет равно w ($1 \leq w \leq 10^4$).
- 3 u v — запрос кратчайшего расстояния от города u до города v . В случае, если города находятся в разных провинциях, длина полагается равной -1 .

Во всех запросах номера городов корректны, то есть $1 \leq u, v \leq N$, где N обозначает общее число городов в государстве.

Формат входных данных

В первой строке входного файла содержится число N — количество городов в государстве. В следующей строке содержится число M — количество запросов. Далее в каждой строке содержится описание запроса. Описание соответствует условию.

$$1 \leq N \leq 50\,000, 1 \leq M \leq 100\,001.$$

Формат выходных данных

Для каждого запроса типа 3 выведите единственное число: ответ на запрос.

Примеры

shortest.in	shortest.out
5 8 2 1 3 10 2 1 2 3 1 1 2 3 1 3 2 1 3 9 2 2 3 1 3 1 5 3 1 3	10 -1 9
5 10 2 2 3 2 2 3 1 2 2 1 5 2 2 5 4 2 2 2 3 2 2 1 5 2 1 3 1 2 3 1 1 2 3 1 2 3 3 1	1